REMARKS

The Examiner maintained the rejection to claim 1 under 35 U.S.C. § 103(a) as being unpatentable over the patent to White in view of the article to Wey in further view of the article to Heikes. Applicant respectfully disagrees.

Claim 1 is directed to a recursive squaring circuit that may be used, for example, in cryptography applications. Claim 1 recites, *inter alia*, an "interoperability" between a host processor, a co-processor, and a hardware logic circuit. Specifically, the host processor, which may be a secure processor, recursively reduces starting integer values to obtain reduced-length integer values for output to the co-processor. The co-processor, which may be an unsecure processor, takes these reduced-length integer values as input, and further recursively reduces them to obtain hardware-length integer values for output to the hardware logic circuit. The hardware logic circuit squares the hardware-length integer values.

Notably, this "interoperability" of claim 1 goes beyond a mere ability to pass integer values between a host-processor and a co-processor via an electrical connection, as the Examiner appears to suggest. Rather, it goes to how the processors operate to obtain the values that *are* passed between the claimed processors and the claimed hardware logic circuit. Because none of the references, alone or in combination, teach or suggest this "interoperability," the §103 rejection of claim 1 fails.

White, for example, discloses a fully-parallel digital circuit that performs mathematical operations on the different parts of a single starting value (i.e., a most significant part, a least significant part, and the sum of the most and least significant parts). However, whatever White teaches with respect to performing mathematical operations on a starting integer value occurs entirely within the physical confines of the disclosed circuit. That is, the White circuit takes an input value to be squared (e.g., an integer), performs parallel mathematical operations on the parts of the integer (i.e., concurrently), and yields a final result. Because the White circuit starts

and finishes the squaring calculation, White does not teach or suggest sending reduced integer values to another processor or hardware logic circuit for further reduction and/or computation. White also does not teach or suggest receiving already-reduced integer values for further reduction or computation.

The Heikes article experiences the same deficiency. Heikes discloses a multiplier array included on a co-processor, and discloses that the co-processor may be part of a PA-RISC processor. Like White, the disclosed multiplier array accepts an input value and outputs a final result. Heikes does not teach or suggest that any values should or could be passed to another circuit for further reduction and/or computation as called out by claim 1. The Examiner cites Heikes because it teaches that the multiplier array is part of a co-processor integrated with a PA-RISC processor. However, this goes only to the physical location of the Heikes multiplier array. It does not mean that Heikes teaches or suggests the "interoperability" of claim 1. Heikes never teaches or suggests the multiplier array receiving reduced integer length values from the PA –RISC processor for further reduction or computation. Heikes also never teaches or suggests recursively reducing starting integer values, and sending the reduced values to another processor or hardware logic circuit for further reduction and/or computation.

Finally, Wey discloses a multiplier circuit that recursively partitions a multiplier and a multiplicand into a fixed number of groups having a fixed number of bits each. However, as the Examiner admits in the Office Action, the Wey multiplier circuit performs the calculations (i.e., summation) on the decomposed parts. That is, the Wey circuit receives an input value, reduces the values, and sums the parts to yield a final result. Wey never teaches or suggests sending the reduced values to another processor or hardware logic circuit for further reduction and/or computation. Wey also does not teach or suggest that the disclosed multiplier circuit receives and operates on already-reduced multipliers and multiplicands.

Respectfully, the rejection appears based on a collection of independent concepts pieced together using claim 1 as a blueprint, and thus, equates to impermissible hindsight reconstruction. None of the references teaches or suggests the interoperability of claim 1 because each reference discloses a circuit that performs its respective mathematical operations to completion independently of other processors or hardware logic circuits. Because none of the cited references teaches or suggests this interoperability of claim 1, they necessarily cannot be combined to teach or suggest the interoperability of claim 1. As such, the §103 rejection fails as a matter of law.

The Examiner also rejected claims 9, 17, 27, 29, 31, and 33 under 35 U.S.C. § 103(a) citing the same references and similar reasons to those stated above, Claim 9 however, contains language similar to that of claim 1. Thus, for the reasons stated above, none of the references teaches or suggests, alone or in combination, claim 9.

Claims 17 and 27 each recite a host processor that recursively reduces starting integer values, and a co-processor that performs mathematical calculations on the output of the host processor. Thus, claims 17 and 27 are patentably non-obvious over the cited references for reasons similar to those stated above. Additionally, however, claims 17 and 27 also recite randomly ordering the output of the host processor prior to passing the values to the co-processor. However, each of the references performs their respective operations from start to finish. Thus, there is no need for the references to teach or suggest randomly order values that they never pass. This fact is also evidenced in the Office Action. Particularly, the Examiner never asserts that the references teach or suggest randomly ordering integer values. The only mention of this aspect of the claimed invention in the Office Action is the Examiner's admission that White fails to teach or suggest randomly ordering the claimed ending integer values. Simply put, none of the cited references teaches or suggests, alone or in combination, claim 17 or 27.

Claims 29 and 31 contain language similar to that of claims 17 and 27, but explicitly identify the host processor as a secure processor. Therefore, in addition to those reasons stated above, the §103 rejection of claims 29 and 31 also fails because none of the references teaches or suggests a secure processor.

Finally, claim 33 is directed to a method of designing a logic circuit. The Examiner supports the rejection of claim 33 simply by stating, "... a square is a special case of multiplication wherein the two values being multiplied are equal." This says nothing about any of the elements of claim 33. Claim 33 recites, *inter alia*, defining index values, recursive logic circuits, and base logic circuits. It recites nothing about squares and multiplication. To establish a *legally sufficient* §103 rejection, the Examiner must show, *inter alia*, that the relied-upon references teach or suggest each element of the rejected claim. In this case, the rejection fails to address any of the elements of claim 33. As such, the §103 rejection to claim 33 fails as a matter of law.

Therefore, for the reasons stated above, none of White, Wey, and Heikes teaches or suggests, alone or in combination, any of the pending claims. Accordingly, Applicant respectfully requests allowance of all pending claims.

Respectfully submitted,

COATS & BENNETT, P.L.L.C.

Dated: August 26, 2005

Stephen A. Herrera
Registration No.: 47,642
P.O. Box 5
Raleigh, NC 27602
Telephone: (919) 854-1844